A PUBLIC KEY CRYPTOSYSTEM BASED ON A POLYNOMIAL KNAPSACK

Jae-Gug Bae, Dong-Gyun Kim

Abstract

We introduce a new public key cryptosystem from a polynomial knapsack problem, which is a generalized knapsack problem in a polynomial ring over $\mathbf{Z}$ modulo a fixed polynomial. It's encription and decryption process is very fast. Both take $O(n)$ operations where n is the bit length of a message. Also the security of the system is based on the difficulty of a subset sum problem of high density and the complexity of the operations in a factored polynomial ring.

1 Introduction

Since Diffie and Hellman [3] have introduced the idea of public key cryptography, there has been a lot of efforts and successes in the implementations of public key cryptosystems. At the very beginning, the Merkle-Hellman [9] scheme which used the knapsack problem was suggested. But in 1982, Adam Shamir [11] made the first successful attack on the basic form of the Merkle-Hellman scheme. After that many cryptographer tried to obtain a secure system based on the NP-completeness of the knapsack problem. Most of the knapsack-type PKC have used a hidden super-increasing sequence in the secret key. Brickell [1], Lagarias and Odlyzko [7], Schnorr and others [12] have broken most PKC based on the knapsack problem successively. One of the major attacks was a "low density" attack which used the lattice basis reduction algorithm. By now, only few knapsack-type PKC which include Chor-Rivest scheme [2] are survived against the lattice attack. (See [13] also.)

In this paper, we give another try of using the knapsack problem for our new cryptosystem. Our system is mainly different from others in using several batches of super-increasing sequences instead of just one sequence so that one can increase the density of the public key high enough. To use the polynomial ring over $\mathbf{Z}$ modulo a fixed private polynomial Q in order to conceal the set of super-increasing sequences is also a central characteristic of our system.

2 The Proposed Cryptosystem

In this section, we describe our new public key cryptosystem, which is constructed on the polynomial ring $\mathbf{Z}[x]$ modulo an integer M and a fixed polynomial Q. Our secret key will be a set of polynomials with leading coefficients selected from a set of super-increasing sequences and our public key will be constructed by multiplying an invertible polynomial modulo Q to the secret polynomials.

2.1 Setting notations

We choose four positive integers u, v, l, N so that $vl < N$. Let $n = ul$ and $\mathbf{Z}_M = \mathbf{Z}/M\mathbf{Z}$ where M is a positive integer, which will be determined later. Fix a polynomial $Q \in \mathbf{Z}_M[x]$ of degree N and let $R = \mathbf{Z}_M[x]/Q$. An element of R will be written as a polynomial or a vector,

$$F = \sum_{i=0}^{N-1} F_i x^i = (F_0, F_1, \cdots, F_{N-1}) .$$

Also, we will choose l super-increasing sequences of length u and n polynomials in R.

2.2 Generating keys

Choose n polynomials $f_1, f_2, \cdots, f_n$ in R with $f_i = (f_{i0}, f_{i1}, \cdots, f_{i(N-1)})$ $1 \le i \le n$, so that $f_{ij} = 0$ if $i = su + t$ with $0 \le s \le l-1$, $1 \le t \le u$ and $j > N - (s+1)v$. To avoid notational confusion, we use $f(i,j)$ for f_{ij} in parallel. The sets of leading coefficients $\{f(1, N-v), f(2, N-v), \cdots, f(u, N-v)\}$, $\{f(u+1, N-2v), f(u+2, N-2v), \cdots, f(2u, N-$

$2v)\}, \cdots, \{f((l-1)u+1, N-lv), f((l-1)u+2, N-lv), \cdots, f(ul, N-lv)\}$ are supposed to form l super-increasing sequences and M is chosen so that

$$M > \sum_{i=1}^{n} \max\{f(i,j) \mid 0 \leq j \leq N-1\}.$$

Now we take an invertible element $G \in R$ and define $F_i = f_i \cdot G$ for $1 \leq i \leq n$. See the small example below with $u=3, v=2, l=3, N=9$.

$$\begin{array}{lllllllll}
f_1 = (38, & 40, & 28, & 29, & 26, & 48, & 38, & 15, & 0) \\
f_2 = (16, & 51, & 5, & 47, & 43, & 14, & 48, & 18, & 0) \\
f_3 = (22, & 33, & 9, & 30, & 34, & 44, & 16, & 34, & 0) \\
f_4 = (15, & 34, & 47, & 17, & 37, & 8, & 0, & 0, & 0) \\
f_5 = (15, & 27, & 14, & 12, & 36, & 9, & 0, & 0, & 0) \\
f_6 = (0, & 19, & 2, & 49, & 32, & 19, & 0, & 0, & 0) \\
f_7 = (11, & 16, & 23, & 13, & 0, & 0, & 0, & 0, & 0) \\
f_8 = (40, & 2, & 23, & 15, & 0, & 0, & 0, & 0, & 0) \\
f_9 = (7, & 23, & 42, & 31, & 0, & 0, & 0, & 0, & 0)
\end{array}$$

$$\begin{array}{lllllllll}
F_1 = (626, & 670, & 326, & 207, & 663, & 235, & 580, & 625, & 89) \\
F_2 = (341, & 532, & 657, & 2, & 134, & 185, & 417, & 357, & 201) \\
F_3 = (387, & 234, & 40, & 558, & 78, & 43, & 329, & 370, & 44) \\
F_4 = (313, & 602, & 95, & 352, & 99, & 659, & 485, & 181, & 334) \\
F_5 = (568, & 601, & 613, & 197, & 167, & 412, & 128, & 317, & 4) \\
F_6 = (153, & 108, & 149, & 243, & 344, & 115, & 618, & 436, & 473) \\
F_7 = (38, & 155, & 216, & 146, & 205, & 171, & 190, & 424, & 136) \\
F_8 = (152, & 585, & 262, & 616, & 70, & 670, & 553, & 127, & 168) \\
F_9 = (135, & 5, & 216, & 638, & 153, & 292, & 447, & 346, & 532)
\end{array}$$

Here we took $G = (230, 372, 56, 202, 235, 117, 565, 5, 614)$ and $Q = (611, 344, 458, 514, 146, 24, 143, 430, 256, 1)$. Note that the sets of leading coefficients $\{15, 18, 34\}$, $\{8, 9, 19\}$, $\{13, 15, 31\}$ form three super-increasing sequences.

[Public Key] The integer M and polynomials $F_1, F_2, \cdots, F_n$
[secret Key] Polynomials G, G^{-1}, Q and $f_1, f_2, \cdots, f_n$

2.3 Encryption and Decryption

Let $m = (m_1, m_2, \cdots, m_n)$ be a message where each $m_i \in \{0, 1, x, x^2, \cdots, x^{v-1}\}$. Then the encrypted message e would be the polynomial

$$e \equiv \sum_{i=1}^{n} m_i F_i \pmod{M}.$$

We describe the decryption.
[I] First of all, calculate

$$s_1 = e \cdot G^{-1} = \sum_{i=1}^{n} m_i f_i = (s(1,0), s(1,1), s(1,2), \cdots, s(1, N-1))$$

in the ring R and then solve a super-increasing knapsack problem

$$\sum_{i=1}^{u} x_i f(i, N-v) = s(1, N-1).$$

Let $(\delta_{1\,1}, \delta_{1\,2}, \cdots, \delta_{1\,u})$ be the solution. Next, we calculate

$$s_2 = s_1 - x^{v-1} \sum_{i=1}^{u} \delta_{1\,i} f_i = (s(2,0), s(2,1), \cdots, s(2, N-2), 0)$$

and solve $\sum_{i=1}^{u} x_i f(i, N-v) = s(2, N-2)$ to obtain the solution $(\delta_{2\,1}, \delta_{2\,2}, \cdots, \delta_{2\,u})$ and we put

$$s_3 = s_2 - x^{v-2} \sum_{i=1}^{u} \delta_{2\,i} f_i = (s(3,0), s(3,1), \cdots, s(3, N-3), 0, 0).$$

Repeating this process v times, we have

$$s_{v+1} = s_v - \sum_{i=1}^{u} \delta_{v\,i} f_i = (s(v+1,0), s(v+1,1), \cdots, s(v+1, N-v-1), 0, \cdots, 0)$$

and coclude $(m_1, m_2, \cdots, m_u) = \sum_{i=1}^{v} x^{v-i} (\delta_{i\,1}, \delta_{i\,2}, \cdots, \delta_{i\,u})$.

[II] For the next batch $(m_{u+1}, m_{u+2}, \cdots, m_{2u})$, observe that $s_{v+1} = \sum_{i=u+1}^{n} m_i f_i$ and perform exactly the same procedure of [I]. Invoking step [I] l times, we obtain original message $m = (m_1, m_2, \cdots, m_n)$.

3 Parameter Selection and Efficiency

3.1 Parameter selection

For the secure and efficient cryptosystem, we need to choose parameters carefully. Comparing the coefficients of an encrypted message

$$e \equiv \sum_{i=1}^{n} m_i F_i \pmod{M},$$

we have N (almost linear) equations that one can analyse. Thus we must take N small compared to n. Because $N > vl$, v and l must be small also. In practical use, we will take $v \leq 10, l \leq 30$ so that $N = vl + k \leq 40$ with $k \leq 10$. To avoid a brute force attak on a message, we must have quite large n. We will have $100 \leq n \leq 1000$. Since $n = ul$, after determining l first, one can choose u so that n is appropriate.
For the selection of l super-increasing sequences of length u, we choose a moderately small number randomly and denote it by a_1. If $a_1, a_2, \ldots, a_i$ are chosen inductively, then we take a random integer $r \in \{1, 2, 3, \ldots, 10\}$ and let $a_{i+1} = \sum_{j=1}^{i} a_j + r$.

3.2 Efficiency comparison

In this section, we examine the efficiency of our system. Given input message parameter of bit length n, the encryption and decryption speeds are both $O(n)$, though the public and private key sizes are both $O(n^2)$. The message expansion rate varies upon variables u, v and l. The precise rates is

$$\frac{v \cdot (u + \log_2 l)}{u \cdot \log_2(v+1)}.$$

Therefore it is recommended to take $v = 1$ to reduce a message expansion rate. (See section 3.3.) The following table compares main characteristics of RSA [10], McEliece [8], GGH [4], NTRU [5], and the Polynomial Knapsack Cryptosystem where the number n represents the length of a message parameter.

	Polynomial Knapsack	NTRU	RSA	McEliece	GGH
Encryption Speed	n	n^2	n^2	n^2	n^2
Decryption Speed	n	n^2	n^3	n^2	n^2
Public Key	n^2	n	n	n^2	n^2
Private Key	n^2	n	n	n^2	n^2
Message Expansion	varies	varies	$1-1$	$2-1$	$1-1$

3.3 Practical Implementation

We present four examples of practical implementations with suitable choices of parameters. In all examples, the first elements of super-increasing sequences are chosen between 10 and 20, randomly. For given public polynomials $F_1, F_2, \cdots, F_n$, we define the density

$$\delta(F_1, F_2, \cdots, F_n) = \frac{n}{\max\{\log_2 F(i,j) | 1 \le i \le n, 0 \le j \le N-1\}}$$

[Example 1]

$(v, u, l, n, N) = (1, 25, 6, 150, 9)$
Public Key $= 2^{15}$ bits Secret Key $= 2^{13}$ bits
Density= 5 Message Expansion Ratio = 1.8

[Example 2]

$(v, u, l, n, N) = (1, 15, 18, 270, 21)$
Public Key $= 2^{17}$ bits Secret Key $= 2^{15}$ bits
Density= 12 Message Expansion Ratio = 1.6

[Example 3]

$(v, u, l, n, N) = (1, 23, 20, 460, 24)$
Public Key $= 2^{18}$ bits Secret Key $= 2^{16}$ bits
Density= 15 Message Expansion Ratio = 1.5

[Example 4]

$(v, u, l, n, N) = (3, 70, 8, 560, 27)$
Public Key $= 2^{20}$ bits Secret Key $= 2^{17}$ bits
Density= 7 Message Expansion Ratio = 1.9

4 Security Analysis

In this section we examine some possible attacks on the cryptosystem. The lattice attack based on LLL algorithm will be a major one.

4.1 Brute force attack

Trying all ppssible $G^{-1}, Q \in \mathbf{Z}_M[x]$ of degree $N-1$, N, respectively, and testing if $F_i \cdot G^{-1}$ $(1 \leq i \leq n)$ have very special forms like our secret key f_i, one may recover the secret key. But in this case an attacker will have M^{2N-1} choices. This is much worse than the message attack which has $(v+1)^n$ choices. One can avoid these brute attacks by simply increasing the number n.

4.2 Lattice attack

After Lagarias and Odlyzko [7] have devised a lattice attack which is effective against low density knapsacks, many reasearchers improved lattice basis reduction algorithm from which originated that of Lenstra, Lenstra and Lovász [6]. In our specific case, one can use LLL algorithm by considering $\{0,1\}$-knapsack problem of $v \cdot n$ polynomials $F_1, F_2, \cdots, F_n$, $xF_1, xF_2, \cdots,$ $xF_n, \cdots, x^{v-1}F_1, x^{v-1}F_2, \cdots, x_{v-1}F_n$. For the notational simplicity, let us assume that $v = 1$. As it is noted is in [2], a simple application of LLL attack does not work due to the high density of public key. As a method of reducing density, one may form the following lattice L;

$$\begin{pmatrix} 1 & 0 & \cdots & 0 & \sum_{j=0}^{N-1} c_j F(1,j) \\ 0 & 1 & \cdots & 0 & \sum_{j=0}^{N-1} c_j F(2,j) \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & \sum_{j=0}^{N-1} c_j F(n,j) \\ 0 & 0 & \cdots & 0 & -\sum_{j=0}^{N-1} c_j s_j \end{pmatrix}$$

for a given polynomial knapsack problem

$$\sum_{i=1}^{n} \varepsilon_i F_i = (s_0, s_1, \cdots, s_{N-1}), \quad \varepsilon_i \in \{0,1\}.$$

Then L contains the vector $(\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n)$ which is comparatively small. Let $a_i = \sum_{j=0}^{N-1} c_j F(i,j)$. By taking c_j's arbitrarily large, one can reduce the density of $a_1, a_2, \cdots, a_n$ expecting that LLL algorithm works efficiently for L. Saying on experimental base, this method works brilliantly for small n such as $n \leq 40$. But for $n \geq 100$, the algorithm fails to find the solution vector even if the density of $\{a_i | 1 \leq i \leq n\}$ is less that 0.01. It seems that this phenomena results from the non-randomness of $\{a_i | 1 \leq i \leq n\}$. We suspect this is a virgin territory that needs further research.

References

[1] E.F. Brickell, *Breaking iterated knapsacks*, Advances inCryptology: Proceedings of Crypto'84, G.R. Blakely and D. Chaum eds., Springer-Verlag (1985), 342–358.

[2] B. Chor, R.L. Rivest, *A knapsack-type public key cryptosystem based on arithmetic in finite fields*, IEEE Trans. on Information Theory **IT-34** (1988), 901–909.

[3] W. Diffie, M.E. Hellman, *New direction in cryptography*, IEEE Trans. on Information Theory **22** (1976), 644–654.

[4] O. Goldreich, S. Goldwasser, S. Halevi, *Public-key cryptosystems from lattice reduction problems*, MIT, Laboratory for Computer Science, preprint, November 1996.

[5] J. Hoffstein, J. Pipher, J.H. Silverman, *NTRU: A ring based public key cryptosystem*, Algorithmic Number Theory, J.P. Buhler eds., Lecture Notes in Computer Science **1423** (1998), 267–288.

[6] A.K. Lenstra, H.W. Lenstra, L. Lovász, *Factoring polynomials with rational coefficients*, Math. Ann. **261** (1982), 515–534.

[7] J.C. Lagarias, A.M. Odlyzko, *Solving low density subset sum problems*, Proc. 25th Annual IEEE Symposium on Foundations of Computer Science (1983), 1–10.

[8] R.J. McEliece, *A public-key cryptosystem based on algebraic coding theory*, JPL Pasadena, DSN Progress Reports **42–44** (1978), 114–116.

[9] R.C. Merkle, M. Hellman, *Hiding information and signatures in trapdoor knapsacks*, IEEE Trans. on Information Theory **IT-24** (1978), 525–530.

[10] R.L. Rivest, A. Shamir, L. Adleman, *A method for obtaining digital signatures and public key cryptosystems*, Communications of the ACM **21** (1978), 120–126.

[11] A. Shamir, *A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem*, Proc. 23rd Annual IEEE Symposium on Foundations of Computer Science (1982), 145–152.

[12] C.P. Schnorr, M. Euchner, *Lattice basis reduction: improved practical algorithms and solving subset sum problems*, Mathematical Programing **66** (1994), 181–199.

[13] C.P. Schnorr, H.H. Hoerner, *Attacking the Chor Rivest cryptosystem by improved lattice reduction*, Proc. EUROCRYPT 1995, Lecture notes in Computer Science 921, Springer-Verlag, 1995, 1–12.